集成学习笔记

宋佳欢

2019 年 9 月 12 日

目录

1 Bagging **1**
1.1 随机森林 1
2 Boosting **1**
2.1 Boosting 的框架 2
2.2 Adaboost 2

1 Bagging

什么情况适合做 Bagging：模型容易过拟合（很强的模型），降低模型预测的方差。Bagging 并不能帮助模型更好地拟合训练数据。

1.1 随机森林

决策树在训练数据上很容易达到 0 错误率（每个叶节点只包含一个样本）。随机森林时决策树的 Bagging 算法, 两种实现方式：

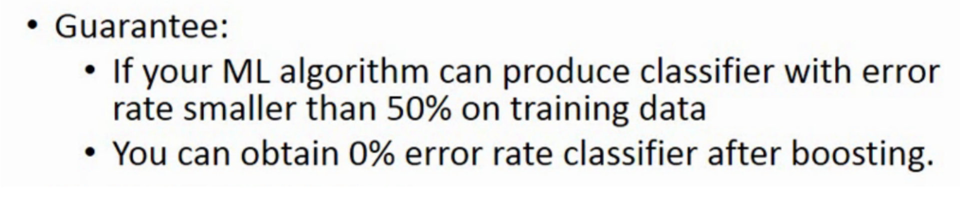
1. 在 m 个样本的数据集 D 中，有放回地随机采样 m 个样本，将其拷贝进 D_1。D_1 中会有重复的样本，且部分 D 中的样本没有在 D_1 中出现。重复上述过程，得到 n 个不同的数据集 $D_1, D_2, \cdots, D_n$，用这些数据集分别训练 n 个决策树，构成随机森林。

2. 随机地限制某些特征在生成决策树时禁用，这样也能得到若干不同的决策树。

out-of-bag validation for Bagging：若方法 1 训练决策树，可将那些在训练集中没有出现过的那些样本用于测试，这样就可以不用划分训练集和测试集了。

2 Boosting

什么情况适合做 boosting：想要提升弱分类器的分类性能



2.1 Boosting 的框架

1. 得到一个弱分类器 $f_1(x)$。
2. 找到另一个分类器 $f_2(x)$ 取帮助 $f_1(x)$。但是两者不能太相似，最好是互补的。
3. 同理，再找到另一个分类器 $f_3(x)$ 取帮助 $f_2(x)$。

$\cdots$，最后结合所有的分类器。每个分类器都是按顺序学习的。

为了获取不同的分类器，在不同的数据集上训练。为了得到不同的训练集，可以重采样来制造多个数据集；也可以为每个样本设置不同的权重，在实现中，只需在损失函数前为每个样本乘上一个权重即可：

$$L(f)\sum_n l(f(x^n,\hat{y}^n)) \Longrightarrow L(f)\sum_n u^n l(f(x^n,\hat{y}^n))$$

其中 u^n 即为每个样本的权重。

2.2 Adaboost

主要思想：使用分类器 $f_1(x)$ 分类失败的那些样本取训练 $f_2(x)$。

定义 $f_1(x)$ 在训练集上的错误率为 ε_1

$$\varepsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{\sum_n u_1^n}$$

u_1^n 为训练第一个分类器时的第 n 个样本的权重。δ 函数括号内成立时等于 1，否则等于 0。若弱分类器由于随机分类，则有错误率 $\varepsilon_1 < 0.5$。

根据 u_1^n 调整 u_2^n，即增大 $f_1(x)$ 分类错误的样本的权重，减小分类正确的样本的权重，使得在训练 $f_2(x)$ 时，分类器更加侧重于 $f_1(x)$ 分类错误的样本的权重，使得两个分类器互相互补。调整权重使得 u_2^n 满足：

$$\frac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{\sum_n u_2^n}$$